IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Toshiaki Kirihata, et al.

Assignee: SanDisk Corporation

Title: FLASH EEPROM SYSTEM

Serial No.: Not yet assigned Filing Date: Herewith

Docket No.: M-10187-44C US

San Francisco, California
October 30, 2001

BOX PATENT APPLICATION
COMMISSIONER FOR PATENTS
Washington, D. C. 20231

PRELIMINARY AMENDMENT

Dear Sir:

Please amend the accompanying continuation application, concurrently with its filing, as follows:

IN THE SPECIFICATION:

Page 1, between lines 3 and 4, insert the following:

--Cross-Reference to Related Applications

This is a continuation of patent application serial no. 09/280,385, filed March 3, 1999, which is a continuation of patent application serial no. 08/771,708, filed December 20, 1996, now patent no. 5,991,517, which is a continuation of patent application serial no. 08/174,768, filed December 29, 1993, now patent no. 5,602,987, which is a continuation of patent application serial no. 07/963,838, filed October 20, 1992, now patent no. 5,297,148, which in turn is a division of patent application serial no. 07/337,566, filed April 13, 1989, abandoned.--

Rewrite the paragraph at p. 11, ln. 23 - p. 12, ln. 5, to read as follows:

Optimized erase implementations have been disclosed in two copending U.S. patent applications. They are copending U.S. patent applications, Serial No. 204,175, filed June 8, 1988, by Dr. Eliyahou Harari, now patent no. 5,095,344, and one entitled "Multi-State EEprom Read and Write Circuits and Techniques," Serial No. 07/337,579, filed April 13, 1989, now abandoned, by Sanjay Mehrotra and Dr. Eliyahou Harari. The disclosures of the two applications are hereby incorporate by reference. The Flash EEprom cells are erased by applying a pulse of erasing voltage followed by a read to verify if the cells are erased to the "erased" state. If not, further pulsing and verifying are repeated until the cells are verified to be erased. By erasing in this controlled manner, the cells are not subject to over-erasure which tends to age the EEprom device prematurely as well as make the cells harder to program.

Rewrite the paragraph on page 22, lines 8 - 23, to read as follows:

After the bytes for a write cycle have been loaded into the selected memory device, the controller issues a program command to the memory device and initiate a write cycle. Optimized implementations of write operation for Flash EEprom device have been disclosed in two previously cited co-pending U.S. patent applications, Serial No. 204,175, now patent no. 5,095,344, and one entitled "Multi-State EEprom Read and Write Circuits and Techniques," Serial No. 07/337,579, filed April 13, 1989, now abandoned. Relevant portions of the disclosures are hereby incorporated by reference. Briefly, during the write cycle, the controller applies a pulse of programming (or writing) voltages. This is followed by a verify read to determine if all the bits have been programmed properly. If the bits did not verify, the controller repeats the program/verify cycle until all bits are correctly programmed.

Rewrite the paragraph on p. 25, ln. 32 - p. 26, ln 11, to read as follows:

In the present invention, a system of Flash EEprom is used to provide non-volatile memory in place of traditional system memories such as disk storage. However, Flash EEprom memory is subject to wearing out by excessive program/erase cycles. Even with the improved Flash EEprom memory device as disclosed in co-pending U.S. patent applications, Serial No. 204,175, now patent no. 5,095,344, and one entitled "Multi-state EEprom Read and Write Circuits and Techniques," by Sanjay Mehrotra and Dr. Eliyahou Harari, Serial No. 07/337,579, filed April 13, 1989, now abandoned, the

endurance limit is approximately 10^6 program/erase cycles. In a ten-year projected life time of the device, this translates to a limit of one program/erase cycle per 5 minutes. This may be marginal in normal computer usage.

IN THE CLAIMS:

Please cancel the original parent application claims 1-62, without prejudice, and substitute the following new claims therefore:

--63. A defect management engine comprising:

a memory array comprising a plurality of groups of data cells and address cells, said cells in each of said groups of data cells and address cells respectively storing data and addresses;

means for accessing at least one of said plurality of groups of data cells and address cells;

means for detecting an address match condition for each of said accessed groups, wherein the address of said accessed group matches an inputted address; and

means for outputting data from said accessed group of data cells when said address match condition is detected.

64. The defect management engine as recited in claim 63, further comprising at least one main memory wherein said data cells are redundancy data cells for replacing defective ones of said data cells in said at least one main memory, and said address cells are redundancy address cells for storing addresses of defective ones of data cells in said at least one main memory.

65. The defect management engine as recited in claim 63 further comprising means for overriding said accessed group of data cells respectively with new data, when said address match condition is detected

66. The defect management engine as recited in claim 63, wherein said data cells and said address cells are of a same cell type.

67. The defect management engine as recited in claim 63, wherein addresses stored in said address cells are non-volatile.

68. The defect management engine as recited in claim 63 further comprising means for programming addresses stored in said address cells.

69. The defect management engine recited in claim 64, wherein said redundancy data cells are of a same cell type as said data cells in said at least one main memory

70. The defect management engine as recited in claim 68, wherein said means for programming is enabled by a command generated by a controller means.

71. The defect management engine as recited in claim 68, wherein said means for programming replicates said addresses into said address cells by sequentially activating each of said address cells.

72. The defect management engine as recited in claim 68, wherein said means for programming replicates said addresses into said address cells by simultaneously activating at least two of said address cells.

73. The defect management engine as recited in claim 64 further comprising means for enabling a single-bit and a multi-bit redundancy replacement within said at least one main memory.

74. The defect management engine as recited in claim 73, wherein said multi-bit size redundancies replace multiple defective ones of said data cells within said at least one main memory with one group of said plurality of groups of redundancy data cells

75. The defect management engine as recited in claim 73 further comprising means for enabling a variable bit size redundancy replacement in said at least one main memory.

76. The defect management engine as recited in claim 64, wherein said data cells are redundancy data cells for replacing defective ones of said data cells present in at least one domain within said at least one main memory, and said address cells are redundancy address cells for addressing defective ones of said data cells within said at least one domain.

77. The defect management engine as recited in claim 76, wherein said plurality of groups of redundancy data cells and redundancy address cells is assigned to said at least one domain.

78. The defect management engine as recited in claim 76, wherein said at least one domain within said at least one main memory is supported by at least one of said plurality of groups of redundancy data cells and redundancy address cells.

79. The defect management engine as recited in claim 63, wherein said means for accessing at least one of said plurality of groups includes wordline drivers.

80. The defect management engine as recited in claim 79, wherein said means for accessing at least one of said plurality of groups further includes sense amplifiers.

81. The defect management engine as recited in claim 79, wherein said wordline driver enables means for assigning to said plurality of groups of data cells and address cells respective redundancy ones of said redundancy data cells and redundancy address cells to repair a plurality of faults in at least one of said domains.

82. A defect management engine coupled to at least one main memory comprising:

a memory array comprising a plurality of groups of redundancy data cells and redundancy address cells, said cells in each of said groups of redundancy data cells and redundancy address cells respectively storing redundancy data and redundancy addresses;

means for accessing at least one of said plurality of groups of redundancy data cells and redundancy address cells;

means for detecting a redundancy address match condition for each of said accessed groups, wherein the redundancy address of said accessed group matches one of a plurality of inputted addresses; and

means for outputting redundancy data from said accessed group of redundancy data cells and redundancy address cells when said redundancy address match condition is detected.

83. The defect management engine as recited in claim 82 further comprising means for overriding said accessed group of redundancy data cells with new data when said redundancy address match condition is detected.

84. The defect management engine as recited in claim 82, wherein said redundancy data cells and said redundancy address cells are of a same cell type

85. A defect management system comprising a plurality of memory chips, and at least one defect management engine chip coupled to each of said memory chips, said at least one defect management engine chip comprising:

a memory array comprising a plurality of groups of data cells and address cells, said cells in each of said groups of data cells and address cells respectively storing data and addresses;

means for accessing at least one of said plurality of groups of data cells and address cells;

means for detecting an address match condition for each of said accessed groups, wherein the address of said accessed group matches an inputted address; and

means for outputting data from said accessed group of data cells when said address match condition is detected.

86. The defect management system as recited in claim 85 further comprising a non-volatile random access memory chip coupled to each of said memory chips.

87. A defect management system comprising a plurality of chips, each of said chips comprising at least one memory, and at least one defect management engine coupled to said at least one memory, said at least one defect management engine comprising: a memory array comprising a plurality of groups of data cells and address cells, said cells in each of said groups of data cells and address cells respectively storing data and addresses; means for accessing at least one of said plurality of groups of data cells and address cells; means for detecting an address match condition for each of said accessed groups, wherein the address of said accessed group matches an inputted address; and means for outputting data from said accessed group of data cells when said address match condition is detected.

88. The defect management system as recited in claim 87 further comprising at least one non-volatile random access memory coupled to said at least one memory and to said at least one defect management engine.

89. A defect management system comprising a plurality of chips, each of said chips comprising at least one memory, at least one non-volatile random access memory coupled to said at least one memory, and at least one defect management engine coupled to said at least one memory, said at least one defect management engine comprising: a memory array comprising a plurality of groups of data cells and address cells said cells, in each of said groups of data cells and address cells respectively storing data and addresses; means for accessing at least one of said plurality of groups of data cells and address cells; means for detecting an address match condition for each of said accessed groups, wherein the address of said accessed group matches an inputted address; and means for outputting data from said accessed group of data cells when said address match condition is detected.

90. The defect management system as recited in claim 89 further comprising at least one defect management engine chip coupled to each of said chips comprising said at least one memory and to said at least one non-volatile random access memory.

91. A method of managing defects comprising the steps of:

configuring a memory array in a plurality of groups of data cells and address cells, said cells in each of said groups of data cells and address cells respectively storing data and addresses;

accessing at least one of said plurality of groups of data cells and address cells;

detecting an address match condition for each of said accessed groups, wherein the address of said accessed group matches one of a plurality of inputted addresses; and

outputting data from said accessed group of data cells and address cells when said address match condition is detected.

92. The method of managing defects as recited in claim 91 further comprising the step of overriding with new data said accessed group of data cells when said address match condition is detected.

93. The method of managing defects as recited in claim 91, wherein said data cells and said address cells are of a same cell type.

94. The method of managing defects as recited in claim 91, wherein the addresses stored in said address cells are non-volatile.

95. The method of managing defects as recited in claim 91 further comprising means for programming said addresses stored in at least one of said address cells.

96. A method of managing defects comprising the steps of:

configuring a memory array into a plurality of groups of redundancy data cells and redundancy address cells, said cells respectively storing redundancy data and redundancy addresses;

accessing at least one of said plurality of groups of redundancy data cells and redundancy address cells;

detecting a redundancy address match condition for each of said accessed groups, wherein the redundancy address of said accessed group of redundancy address cells matches one of a plurality of inputted addresses; and

outputting redundancy data from said accessed group of redundancy data cells and redundancy address cells when said redundancy address match condition is detected.

97. The defect management engine as recited in claim 96 further comprising the step of overriding with new data said accessed group of redundancy data cells when said redundancy address match condition is detected.--

REMARKS

By this Preliminary Amendment, the original parent application claims are being canceled and a new set of claims being substituted. New claims 63-97 are respectively copies of claims 1-3, 5, 7-9, 14-16, 18, 20-21, 25-27, 31-35, 37, 39-46, 48 and 50-53 of U.S. patent no. 6,141,267 - Kirihata *et al.* (2000). The new claims are directed to defect management, such as described in the "Defect Mapping" section of the present application beginning on page 14, line 13. A copy of the Kirihata *et al.* patent is being filed with this Amendment for the convenience of the Examiner.

An early examination and allowance of the present application are solicited.

EXPRESS MAIL LABEL NO:

EL 873331478 US

Respectfully submitted,

Michael G. Cleveland
Attorney for Applicant(s)
Reg. No. 46,030

DETAILS OF CHANGES TO THE SPECIFICATION TEXT

Page 11, line 23 - page 12, line 5:

Optimized erase implementations have been disclosed in two copending U.S. patent applications. They are copending U.S. patent applications, Serial No. 204,175, filed June 8, 1988, by Dr. Eliyahou Harari, now patent no. 5,095,344, and one entitled "Multi-State EEprom Read and Write Circuits and Techniques," [filed on the same day as the present application] Serial No. 07/337,579, filed April 13, 1989, now abandoned, by Sanjay Mehrotra and Dr. Eliyahou Harari. The disclosures of the two applications are hereby incorporate by reference. The Flash EEprom cells are erased by applying a pulse of erasing voltage followed by a read to verify if the cells are erased to the "erased" state. If not, further pulsing and verifying are repeated until the cells are verified to be erased. By erasing in this controlled manner, the cells are not subject to over-erasure which tends to age the EEprom device prematurely as well as make the cells harder to program.

Page 22, lines 8 - 23:

After the bytes for a write cycle have been loaded into the selected memory device, the controller issues a program command to the memory device and initiate a write cycle. Optimized implementations of write operation for Flash EEprom device have been disclosed in two previously cited co-pending U.S. patent applications, Serial No. 204,175, now patent no. 5,095,344, and one entitled "Multi-State EEprom Read and Write Circuits and Techniques," Serial No. 07/337,579, filed April 13, 1989, now abandoned. Relevant portions of the disclosures are hereby incorporated by reference. Briefly, during the write cycle, the controller applies a pulse of programming (or writing) voltages. This is followed by a verify read to determine if all the bits have been programmed properly. If the bits did not verify, the controller repeats the program/verify cycle until all bits are correctly programmed.

Page 25, line 32 - page 26, line 11:

In the present invention, a system of Flash EEprom is used to provide non-volatile memory in place of traditional system memories such as disk storage. However, Flash EEprom memory is subject to wearing out by excessive program/erase cycles. Even with the improved Flash EEprom memory device as disclosed in co-pending U.S. patent applications, Serial No. 204,175, now patent no. 5,095,344, and one entitled "Multi-state EEprom Read and Write Circuits and Techniques," by Sanjay Mehrotra and Dr. Eliyahou Harari, Serial No. 07/337,579, filed April 13, 1989, now abandoned, [filed on the same day as the present application,] the endurance limit is approximately 10^6 program/erase cycles. In a ten-year projected life time of the device, this translates to a limit of one program/erase cycle per 5 minutes. This may be marginal in normal computer usage.